FUNCTIONS (EXPLANATION):

1) Function: convert_pdf_to_images(input_folder, output_folder, dpi=900)

   This function converts PDF files to images and stores them in the specified output folder.It takes the input_folder as the main parent folder containing PDF files and output_folder as the folder where images will be stored. dpi is the resolution in dots per inch, and by default, it is set to 900.

2) Function: has_data(chunk)

   This function checks if a given image chunk (cropped portion) has any relevant data.It processes the chunk to determine if it contains any contours or lines.It returns True if the chunk contains significant information, otherwise False.

3) Function: crop_image_into_chunks(image_path)

   This function takes an image path as input and divides the image into smaller chunks for better readability.It first opens the image using PIL, then divides the image into m x n chunks, where m and n are defined as 15 in this code.Padding is added to each chunk to ensure that the entire content is captured.Chunks are then created and returned as a list.

4) Function: process_image(image_path)
   This function processes each image chunk and extracts colors and numbers from the text using EasyOCR. The extracted colors and numbers are returned as a tuple.

5) Function: process_subfolder(input_subfolder_path, output_subfolder_path)

   This function processes each subfolder (which contains image chunks) within the input folder. It creates an Excel file for each subfolder containing color and number information extracted from the image chunks.

6) Function: process_folder(input_folder, output_folder)

   This function replicates the input directory structure in the output folder and processes each subfolder. For each subfolder, it calls process_subfolder() to generate Excel files containing color and number information.

OVERALL CODE FUNCTIONING:

The code starts by taking the user's input for the input_folder (parent folder with PDFs) and output_folder (intermediate folder for chunks and images).

It then calls convert_pdf_to_images() to convert all the PDF files in the input_folder to images and store them in the output_folder.

Afterward, the code walks through the output_folder and divides the images into chunks using crop_image_into_chunks(). Each chunk is saved as a separate image file.

The original images are removed from the output_folder after chunking is complete.

The EasyOCR model is loaded, and the code proceeds to process each chunk, extract color and number information using process_image().

The process_subfolder() function is called to generate Excel files for each subfolder containing color and number information extracted from the image chunks.

Finally, the intermediate folder containing all the chunks and images is removed using shutil.rmtree().

Note: The code uses multithreading in the process_subfolder() function to process multiple chunks concurrently, which can speed up the processing of a large number of chunks. The concurrent.futures module is used for this purpose.